Indexing Iris Database Using Multi-Dimensional R-Trees

Tithy Sahu

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

Indexing Iris Database Using Multi-Dimensional R-Trees

Thesis submitted in

May 2012

to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Tithy Sahu

[Roll No. 108CS030]

under the guidance of

Prof. Banshidhar Majhi

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

Department of Computer Science & Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India. www.nitrkl.ac.in

Dr. Banshidhar Majhi
Professor

May 14, 2012

Certificate

This is to certify that the work in the thesis entitled ***Indexing Iris Database using Multi-Dimensional R-Trees*** by ***Tithy Sahu***, bearing Roll No. **108CS030**, is a record of an original research work carried out by her under my supervision and guidance in partial fulfilment of the requirements for the award of the degree of *Bachelor of Technology* in *Computer Science and Engineering*. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Banshidhar Majhi

Acknowledgment

I would like to express my earnest gratitude to my thesis guide, Prof. Banshidhar Majhi for believing in my ability to work on the challenging domain of biometrics. His profound insights has enriched my research work. The flexibility of work he has offered me has deeply encouraged me producing the research.

My heartfelt thanks to Ms. Hunny Mehrotra for consistently showing me innovative research directions for the entire period of carrying out the research. I am indebted to all the professors, batch mates and friends at National Institute of Technology Rourkela for their cooperation.

I would conclude with my deepest gratitude to my parents, sister and all my loved ones. My full dedication to the work would have not been possible without their blessings and moral support.

Tithy Sahu

Abstract

Iris is one of the most widely used biometric modality for recognition due to its reliability, non-invasive characteristic, speed and performance. The patterns remain stable throughout the lifetime of an individual. Attributable to these advantages, the application of iris biometric is increasingly encouraged by various commercial as well as government agencies. Indexing is done to identify and retrieve a small subset of candidate data from the database of iris data of individuals in order to determine a possible match. Since the database is extremely large, it is necessary to find fast and efficient indexing methods. In this thesis, an efficient local feature based indexing approach is proposed using clustered scale invariant feature transform (SIFT) keypoints, that achieves invariance to similarity transformations, illumination and occlusion. These cluster centers are used to construct R-trees for indexing. This thesis proposes an application of R-trees for iris database indexing. The system is tested using publicly available BATH and CASIA-IrisV4 databases.

Keywords: Biometrics, Iris recognition, SIFT, Indexing, R-trees.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

The term Biometrics refers to the field of development of statistical and mathematical methods applicable to data analysis problems existing in the biological sciences. Biometrics is the science of establishing the identity of an individual based on physiological and behavioural characteristics of the individual.The objective of Biometrics is to promote the use of statistical and mathematical theory towards the development of novel biometrical techniques and their application to new and ongoing subject-matter challenges. Biometric authentication has evolved from the disadvantages of traditional means of authentication. It is more reliable and capable compared to traditional approaches. The problem with token based systems is that the possession could be lost, stolen, forgotten or misplaced. The drawbacks of knowledge based approaches is that it is tough for a person to remember difficult passwords/PINs; while keeping in mind secuirty against attacks.The combination of knowledge and token based system, e.g. automated teller machine (ATM) also cannot satisfy the security requirements. The primary advantage of biometrics over token based and knowledge based approaches is that, it cannot be misplaced, forgotten or stolen. Also it is very difficult to spoof biometric traits of an individual. A generic biometric system operates by taking an input from the user, preprocessing the signal to denoise it to find the region of interest, extracting features, and authenticating an individual based on the result of comparison [2]. Depending upon the application context a biometric system operates in the following modes: enrolment mode, verification mode, identification mode. In

enrolment mode, the feature from a subject is extracted and stored in the database. In verification mode, a subject is authenticated by comparing, one on one, live query biometric template with the database template of the individual whom the subject claims himself to be. In identification mode, the system takes live query template from the subject and searches the entire database to find the best-match template to identify the subject and thereby making it a one-to-many process.

Several biometric traits such as face, iris, fingerprint, voice, face-thermograph, signature are of key research area due to enormous need of security in automated systems. Observing underlying modalities, two basic categories can be identified as: Physiological (or passive) and Behavioral (or active) biometrics [2]. Physiological biometrics are based on measurements or data derived from direct measurement of a human body part. Fingerprint, iris, retina, hand geometry, and face recognition are leading physiological biometrics. Behavioral characteristics, on the other hand, are based on an action taken by a person. Behavioral biometrics are based on measurements of data derived from an action, and thereby indirectly measure characteristics of the human body.A good biometric trait is characterised by use of features that are highly unique, stable, easy to capture,acceptable, collectable and prevents circumvention.

1.1 Iris Biometrics

Iris plays a significant role to provide a promising solution to authenticate an individual using unique texture patterns. It is proved to be the most efficient technique taking reliability and invasiveness into consideration. From the point of view of reliability, the spatial patterns are unique to an individual. From the point of view of invasiveness, iris is protected internal organ whose random texture is stable throughout life. Iris is the most significant feature in the eye image (Figure 1.1). It is in the form of circular ring that contains many interlacing minute characteristics such as freckles, coronas, stripes, furrows, crypts and so on, which are unique to each individual. Pupil is the darkest circular shaped area in the eye image. It controls the amount of light entering the eye by dilation and contraction. Iris is the circular shaped sphincter that separates pupil from the sclera region.

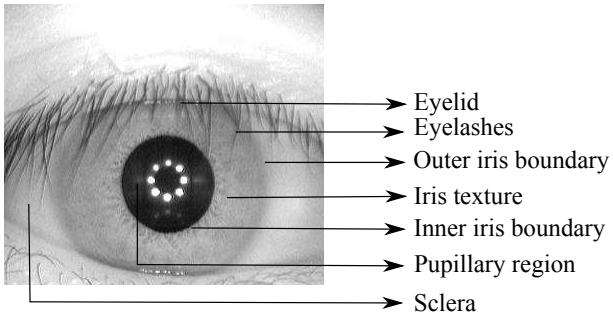


Figure 1.1: Image from CASIA [?] database to depict the anatomy of human eye

Iris, due to its permanence and ease of acquiring, plays a significant role among all the biometric traits. Recent authentication systems need secure, fast and accurate computing for which iris pattern is found to be suitable.In India, a large scale project (Aadhaar) is undertaken to issue Unique Identification (UID) [1] number to each individual across the country using fingerprint and iris. The idea of UID is de-duplication by keeping a check during enrolment that the same citizen is not enrolled more than once.

1.2 Problem Definition

The acquired iris image is localised for inner and outer boundary using image morphology [18]. The annular region between the iris circles is considered for feature extraction. Scale Invariant Feature Transform (SIFT) is applied to iris that provides stable set of features while being less sensitive to local image distortions. Fuzzy c means (fcm) clustering [19] is used to group the number of keypoints, for each iris image, sharing similar descriptor property. These fuzzy cluster centers are used to perform indexing using multidimensional trees. Thus, a local feature based indexing approach is proposed using clustered SIFT keypoints. These cluster centers are used to construct R-trees for indexing. This thesis proposes an application of R-trees for iris database indexing. The system is tested using publicly available BATH and CASIA-Iris-Thousand databases.

1.3 Performance Measures Used

To measure the identification accuracy, each database is divided into two mutually exclusive gallery and probe sets. The gallery set consists of iris templates with known identities. However, probe set consist of iris templates whose identity is to be known. In order to reduce system response time during identification, the gallery set is partitioned into bins. This reduces the number of comparisons required to find the identity of the probe. In this thesis the gallery set is partitioned using multidimensional tree based indexing. Some well known performance measures are used for identification [25].

- **Bin Miss Rate (*bm*):** A bin miss occurs when an attempt is placed in a bin which is not compared with the correct bin for the biometric entity used, and hence will fail to match. The error occurs when the biometric template is placed in the wrong bin during identification.

- **Penetration Rate (*pr*):** During identification, the input feature set is compared to all the templates in the database. Search efficiency is achieved by partitioning the database based on some criteria. Thus, the query template is compared to only selected templates in the appropriate partitions. The portion of total database to be scanned on an average for each search is called penetration coefficient PR, which can be defined by

$$pr = \frac{E}{N} \tag{1.1}$$

  where E is the expected number of comparisons required for single input and N is totla number of comparisons.

- **Gamma (γ):** The choice of an indexing approach becomes crucial when both speed (measured in terms of pr) and accuracy (measured in terms of bm) plays significant role. Thus, γ [20] is found to strike balance between pr and bm. To mark the trade off between pr and bm a new error measure (γ) is used which is defined as

$$\gamma = \sqrt{(1 - pr) * (1 - bm)} \tag{1.2}$$

1.4 Iris Databases

The databases used in all experiments relevant to the research in this thesis are BATH [23] and CASIA-IrisV4 [24]. The proposed system has been tested on two publicly available databases, viz. BATH and CASIA-IrisV4.

BATH Database

Database available from BATH University [23] comprises of images from 50 subjects. For each subject, both left and right iris images are obtained, each containing 20 images of the respective eyes.

CASIA-IrisV4 Database

CASIA-IrisV4 Database is collected by the Chinese Academy of Sciences' Institute of Automation (CASIA). CASIA-IrisV4 is an extension of CASIA-IrisV3 and contains six subsets- CASIA-Iris-Interval, CASIA-Iris-Lamp, CASIA-Iris-Twins, CASIA-Iris-Distance, CASIA-Iris-Thousand, and CASIA-Iris-Syn.[24] CASIA-IrisV4 contains a total of 54,601 iris images from more than 1,800 genuine subjects and 1,000 virtual subjects. All iris images are 8 bit gray-level JPEG files, collected under near infrared illumination or synthesized. In this work, CASIA-Iris-Thousand subset is used. It contains 20,000 iris images from 1,000 subjects. It is the first publicly available iris dataset with one thousand subjects, and hence, is well-suited for studying the uniqueness of iris features and develop novel iris classification and indexing methods [24].

1.5 Thesis Organization

The entire thesis constitutes five chapters following this chapter -

Chapter 2: Literature Review

This chapter outlines the existing work on biometrics indexing, there performance and limitations.

Chapter 3: R-Tree Based Indexing

This chapter explains the data structure proposed by Antonie Guttman [17].

Chapter 4: Applications to Iris Database Indexing

This chapter discusses the entire procedure of the indexing process incorporated to the iris database.

Chapter 5: Results

All the results of the performance measures of the proposed system is shown in this chapter.

Chapter 6: Conclusion

This chapter presents analytical remarks to overall achievements.

Chapter 2

Literature Review

Significant but limited researches for indexing biometric databases exist in literature. A multimodal binning and pruning approach has been proposed using signature and hand geometry database [3]. From experimental results it is observed that the combined system gives penetration rate of 5 %. An indexing approach for hand geometry database is proposed using pyramid technique [4]. The search space is reduced to 8.86% of entire database with 0% false rejection rate (FRR). An efficient indexing technique for large multimodal biometric databases is proposed in [6]. The performance of the proposed system shows that out of 150 query templates, 146 fall in the first match, 147 fall in the top second match and 149 IDs fall in the top 5 matches with 0.66% FRR. Some studies have been made for exclusively indexing iris databases. Hao et al. [7] have proposed iris database indexing approach using beacon guided search (BGS). The proposed system has been tested on real time database collected at UAE and shows substantial reduction in speed with negligible loss in accuracy. The authors in [5] makes use of two approaches for iris indexing. In the first approach, PCA and k-means clustering is used to partition the database. The second approach is based on statistical analysis of pixel intensities and positions of blocks in iris texture. The PCA based indexing gives average penetration of 17% for 80% hit rate, whereas the column-based scheme results in a hit rate of 80% for an average penetration rate of 21%. However, when indexing is based on the block-based statistics of the iris code (8 $\times$ 8 blocks), the average penetration for 80% hit rate is only 8%. The authors in [8]

have proposed an indexing scheme using iris colour for noisy iris images. Results are obtained for change in search range n_{range} and system is performing better for n_{range} = 2. An indexing scheme using energy histogram of DCT subbands is proposed in [9]. This indexing scheme gives considerably low penetration rate of 0.63% with bin miss rate of 43.6%. In [20], from the binary iris image, the n-bit pattern is selected and its locations are searched in iris image using burrows wheeler transform (BWT). The experimental results show significant performance improvement with hit rate of 99.83% at penetration rate of 17.23% and γ (trade-off between hit rate and penetration rate) = 90.90%. In [12], the authors have used the concept of hashing to search the templates. The system is working with penetration rate of approximately 3% to be processed and performance gain of 89.57% with respect to time consumption is obtained.

Local feature based indexing approach is proposed in [13] using geometric hashing of Scale Invariant Feature Transform (SIFT) keypoints.The system is performing with equal penetration rate and bin miss rate of 0.24. An iris colour is used to determine index for reducing the search space [11]. Finally, Speeded Up Robust Features [14] are used for matching query with the retrieved set of iris from the database. Experimental results show that the proposed indexing approach performs with 37.97% penetration rate for 100% hit rate. Recently an effort has been made to further reduce the time required during identification by parallelizing geometric hashing approach [15].

Through asymptotic analysis it has been found that there is significant gain in speed in comparison to traditional geometric hashing approach. From the existing literature it has been studied that the conventional approaches to indexing works using global features. Thus, they are not suitable for iris images taken under unconstrained environment. To develop a robust indexing approach local feature descriptors are used. There are some differences between the proposed approach and the local feature based approaches that exist in literature, these points are highlighted as follows-

- Existing indexing approaches constructs a single multidimensional tree using *global* features whereas the proposed indexing constructs multiple trees (m) using local features.

- There exists an indexing approach suitable for unconstrained iris identification using geometric hashing of SIFT keypoints [13]. The major drawback is that it performs indexing in $O(n^3)$ time.

- A variant of geometric hashing has been developed [15] and performs in $O(n^2)$ time.

An efficient indexing approach is performed using k-d trees constructed using clustered Scale Invariant Feature Transform (SIFT) keypoints. These cluster centers are used to construct k-d trees for indexing. However, there are various limitations of k-d tree based indexing.This approach is static as insertion of new template requires the tree to be re-constructed. Secondly, this indexing approach is incapable of performing de-duplication during insertion of records in the database. Thus, the k-d tree based indexing lacks scalability and fails to keep a check on duplicates. These issues have been addressed using k-d-b tree, which combines the multidimensional capability of k-d tree and balancing efficiency of B tree. However, few limitations are observed during implementation of k-d-b tree based indexing system. These are suitable for point data only. This thesis outlines the use of data structure proposed by Guttman [17] to index the clustered SIFT keypoints. Various performance parameters are measured and compared with existing indexing techniques.

Chapter 3

R-Tree Based Indexing

Antonie Guttman presented a data structure called R-Tree [17] that represents data objects by intervals in several dimensions. In his paper published in 1984 he proposed this dynamic index structure. In this work, an analysis of R-tree is presented to fit the problem of indexing the Iris database. In his paper, Guttman described R-trees as height balanced data structures for multi-dimensional indexing [17]. The data structure is similar to B^+ trees and is used for hierarchical indexing of d-dimensional points represented as d-dimensional minimum bounding rectangles (MBR). The structure is dynamic in nature and does not require periodic re-organization of the index structure. The maximum number of entries for each node is denoted by M and the minimum number is represented by $m \leq \frac{M}{2}$. R-tree has index records in its leaf nodes containing pointers to data objects. The index is completely dynamic. Structure is designed in such a way that a spatial search requires visiting only a small number of nodes. The spatial data is comprised by an MBR which become formated and comprised from an MBR again. This structure continues up to the root. Eventually the root comprise an MBR over all objects. [21] R-trees are, thus, hierarchical data structures based on B^+ trees. They are used for the dynamic organization of a set of n-dimensional geometric objects representing them by the minimum bounding n-dimensional rectangles-MBRs.

Figure 3.1 [22] shows an example of an R-tree.

The figure 3.1 shows a set of the MBRs containing data geometric objects (not

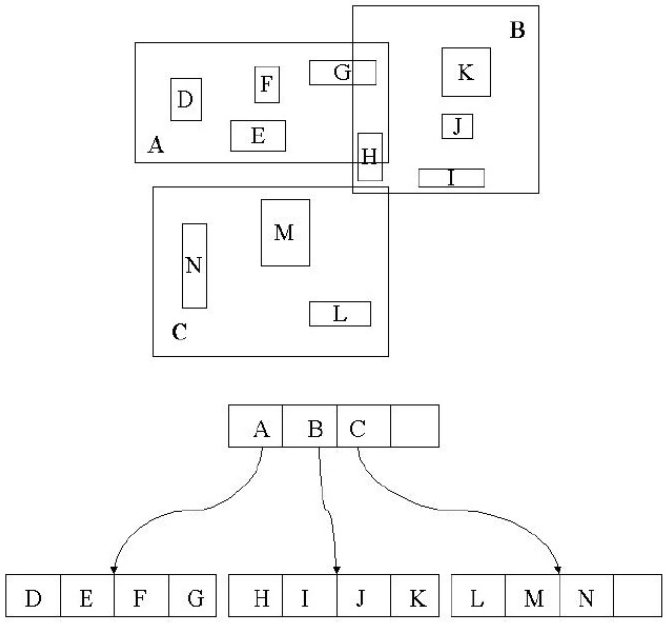


Figure 3.1: R-tree structure

shown). These MBRs are D, E, F, G, H, I, J, K, L, M, and N, which will be stored at the leaf level of the R-tree. It also shows the MBRs (A,B, and C) that organize the aforementioned rectangles into an internal node of the R-tree.

The R-Tree satisfies the following properties [17]:

- Every leaf node contains between m and M index records unless it is the root Thus, the root can have less entries than m. M=maximum number of entries and m=minimum number of entries in one node
- For each index record in a leaf node, I is the smallest rectangle that spatially contains the n-dimensional data object represented by the indicated tuple.
- Every non-leaf node has between m and M children unless it is the root.
- For each entry in a non-leaf node, i is the smallest rectangle that spatially contains the rectangles in the child node.
- The root node has at least two children unless it is a leaf.
- All leaves appear on the same level. That means the tree is balanced.

3.1 Structure of leaf node

Leaf nodes in an R-Tree contain index record entries of the form

(I , tuple-identifier)

where *tuple-identifier* refers to a tuple in the database and I is an n-dimensional rectangle which is the bounding box of the spatial object indexed.

$$\mathrm{I} = (I_0 , I_1 , ..., I_{n-1})$$

Here n is the number of dimensions and I_i is a closed bounded interval *[a,b]* describing the extend of the object along dimension i [17]. Following figure 3.2 [22] represents I as bounding box which contains two records (circle).

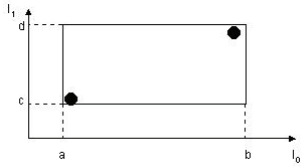


Figure 3.2: Representation of I for n = 2

3.2 Structure of a non-leaf node

The nodes which are no leafs contain entries of the form

(I , child-pointer)

where *child-pointer* is the address of a lower node in the R-Tree and I covers all rectangles in the child node's entries as shown in figure 3.3 [22].

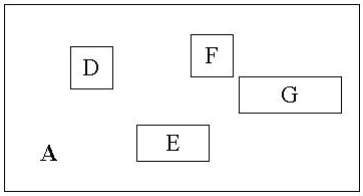


Figure 3.3: Child nodes of A

Chapter 4

Proposed Approach: Applications to Iris Database Indexing

4.1 Preprocessing and Feature extraction

Prior to feature extraction, the acquired iris image is localised for inner and outer boundary using image morphology [18]. The annular region between the iris circles is considered for feature extraction. In order to eliminate noise due to eyelids, sector based approach is used [13] . Due to expansion and contraction of pupil as a natural phenomenon, the texture pattern of iris undergoes linear deformation. Thus, local keypoint features are required that performs for variation in scale along with other transformations. In this thesis, SIFT is applied to iris that provides stable set of features while being less sensitive to local image distortions. The steps involved in feature extraction using SIFT are explained in sequel-

4.1.1 Scale Space Extrema Construction

The keypoints are detected from annular iris image using cascade filtering approach. This is done to achieve invariance to scale. To define the scale space, input iris image (I) is convolved with Gaussian kernel G(x,y,σ) as defined by

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{4.1}$$

where $*$ is the convolution operation,σ defines the width of Gaussian filter and L is the Gaussian smoothed image. The Difference of Gaussian (DOG) images are computed from two nearby scales that differ by constant multiplicative factor

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{4.2}$$

In order to achieve scale invariance, Gaussian blurred images and DOG images are placed together in one octave. The same set of operations are repeated for the next octave by taking downsampled version of the input image.

4.1.2 Keypoint localisation

DOG images are used to detect keypoints with help of local maxima and minima across different scales. Each pixel in DOG image is compared to 8 neighbors in the same scale and 9 neighbors in the scale above and below. The pixel is selected as a candidate keypoint if it is local maxima or minima in 3×3×3 region.

4.1.3 Orientation Assignment

Orientation is assigned to each detected keypoint to achieve invariance to image rotations as descriptor can be represented relative to orientation. To determine key orientation, a gradient orientation histogram is computed in the neighborhood of the keypoint. The scale of keypoint is used to select Gaussian smoothed image L. For each Gaussian smoothed image $L(x, y)$, magnitude $(m(x, y))$ and orientation $(\theta(x, y))$ are computed as

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \tag{4.3}$$

$$\theta(x, y) = \tan^{-1}\left[\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right] \tag{4.4}$$

Orientation histogram is then formed for gradient orientation around each keypoint. The histogram has 36 bins for 360 degree range of orientations and each sample is weighted by gradient magnitude and Gaussian weighted circular window with σ of 1.5 times of scale of keypoint before adding it to histogram. Peaks in the histogram correspond to orientation and any other local peak within 80% of largest peak is used

to create keypoint with the computed orientation. This is done to increase stability during matching [16].

4.1.4 Keypoint descriptor

Once orientation has been selected, the feature descriptor is computed as a set of orientation histograms on 4×4 pixel neighborhoods. The orientation histograms are relative to the keypoint orientation. Histogram contains 8 bins each and each descriptor contains an array of 16 histograms around the keypoint. This generates SIFT feature descriptor of 4×4×8 = 128 elements. The descriptor vector possesses invariance to rotation, scaling, illumination and partial occlusion. For detailed description regarding SIFT the reader is advised to refer Lowe's paper [16].

4.2 Clustering keypoints

The number of keypoints (n) vary across iris images in the database. The traditional approaches to database indexing becomes unsuitable for such local features. This indexing approach is developed using local feature descriptors like SIFT. Fuzzy c means (fcm) clustering [20] is used to group the number of keypoints, for each iris image, sharing similar descriptor property. The approach is based on minimization of objective function which is defined as

$$J_r = \Sigma_{i=1}^{n} \Sigma_{j=1}^{m} u_{ij}^{r} \ (\|x_i - c_j\|)^2 \tag{4.5}$$

where r is any real number greater than 1. Here x_i is k (= 128) dimensional data and n is the number of keypoints to be clustered, m is number of clusters with each cluster centers defined by c_j . u_{ij} is the degree of membership of x_i in c_j . The membership function is updated using

$$u_{ij} = \frac{1}{\Sigma_{l=1}^{m} (\frac{\|x_i - c_j\|}{\|x_i - c_l\|})^{\frac{2}{(r-1)}}} \tag{4.6}$$

For updating cluster centers the following equation is used

$$c_j = \frac{(\Sigma_{i=1}^{n} u_{ij}^{r} \times x_i)}{\Sigma_{i=1}^{n} u_{ij}^{r}} \tag{4.7}$$

The iteration stops when

$$max_{ij}|u_{ij}^{l+1} - u_{ij}^{l}| < \epsilon \tag{4.8}$$

for ϵ between (0,1). The idea is to have transformation from variable number of keypoints (n) to fixed number of clusters (m), ascertained a priori. These fuzzy cluster centers are used to perform indexing using multidimensional trees.

4.3 Indexing

4.3.1 Range Search

As explained in the book [21], given a rectangle, Q, the following query is formed: find all data rectangles that are intersected by Q which is denoted as a range query. For a node entry E, *E.mbr* denotes the corresponding MBR and *E.p* the corresponding pointer to the next level. If the node is a leaf, then *E.p* denotes the corresponding object identifier (*oid*). The following algorithm 1 outlines range search procedure.

Algorithm 1: RangeSearch

Input: R: Tree node, q: Query to be searched

Output: L: List of identifiers

```
1 if R is leaf then
2 |   L ← {e|e ∈ R & e.mbr ∩ q.mbr ≠ φ}
3 else
4 |   S ← {e|e ∈ R & e.mbr ∩ q.mbr ≠ φ}
5 |   foreach e in set S do
6 |   |   RangeSearch(e.mbr, q)
7 |   end
8 end
```

The rectangles that are found by this algorithm constitute the candidates of the filtering step. The actual geometric objects intersected by the query rectangle are found in a refinement step by retrieving the objects of the candidate rectangles and testing their intersection.

Asymptotically, given N = Number of entries in each node, M = Maximum number of entries in each node $\Longrightarrow$ Complexity = O($\log_M n$).

4.3.2 Insertion

Insertions in an R-tree are similar to that in a B^+ tree. The R-tree is traversed to locate an appropriate leaf to accommodate the new entry [21]. The entry is inserted in the found leaf and, then all nodes within the path from the root to that leaf are updated accordingly. In case the found leaf cannot accommodate the new entry because it is full (it already contains M entries), then it is split into two nodes. In case of duplicate entry, which arises when the search rectangle completely overlaps with a leaf node, it is not inserted giving a duplicate entry message. The algorithm 2 shows the steps followed.

Algorithm 2: Insert

Input: *root*: root node, R: Tree node, p: Point to be inserted

```
1  if R is leaf node then
2      if |R| < M then
3          insert p into R
4          update all mbr in path from root to R to cover p.mbr
5          return
6      else
7          R ← split-node(R)
8          Insert(R, p)
9      end
10 else
11     R ← R.id    // child with minimum increase in R.mbr to contain p
12
13     Insert(R, p)
14 end
```

The aforementioned insertion algorithm uses linear split algorithm 3 (linear time

complexity).

Asymptotically, given N = Number of entries in each node, T = Tree height $\Longrightarrow$ Complexity = 2×N×T = O(n). The objective of a split algorithm is to minimize the probability of invoking both created nodes for the same query. The linear split algorithm tries to achieve this objective by minimizing the total area of the two created nodes. Examples of bad and good splits are given in Figure 4.1.

Algorithm 3: Split-node

Input: R: Tree node

Output: w: Updated node after splitting

```
1  Select seeds {e1, e2 | e1, e2 ∈ R} // distance between e1, e2 should be
       maximum compared to other pairs
2
3  Create nodes R1 and R2 using e1 and e2
4  Assign remaining nodes of R to R1 and R2 with minimum increase in mbr
5  if R is root then
6  |   create new root with R1 and R2
7  |   w ← root
8  |   return
9  else
10 |   w ← parent of R
11 |   update R1.mbr and R2.mbr in w
12 |   if |w| > M then
13 |   |   split-node(w)
14 |   end
15 end
```

4.3.3 Deletion

Deletion is performed with the algorithm 4 as given in [21].

Asymptotically, given N = Number of entries in each node, T = Tree height $\Longrightarrow$

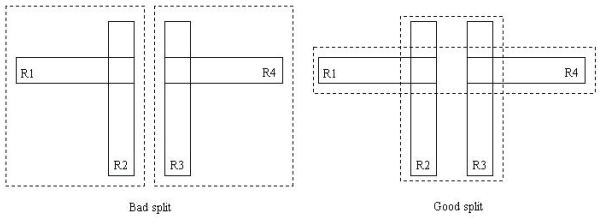


Figure 4.1: Splitting a Node

Algorithm 4: Delete

Input: *root*: root node, R: Tree node, p: Point to be deleted

```
1  if R is leaf then
2  |   RangeSearch(R, p)
3  else
4  |   Find all entries of R that cover p.mbr
5  |   Follow the corresponding subtrees until the leaf L that contains p is found
6  |   Remove p from L
7  end
8  CondenseTree(L)
9  if root has only one child E then
10 |   Remove root
11 |   root ← E
12 end
```

Complexity = 2×N×T = O(n). The handling of an underflowing node (a node with fewer than m entries) is different in the R-tree, as compared to that in B+ tree. Firstly, if a node has an underflow, it is eliminated and inserted again. In a B-Tree, however, the node is fused with an other node. Secondly, the R-Tree is more effcient: Implementation of deletion is easier because Insertion routine can be used. Through the deletion and reinsertion the spatial structure of the tree is incrementally refined.

Algorithm 5: CondenseTree

Input: *root*: root node, L: Leaf node, E: Point to be deleted

Result: Given is the leaf L from which an entry E has been deleted. If after the deletion of E, L has fewer than m entries, then remove entirely leaf L and reinsert all its entries. Updates are propagated upwards and the MBRs in the path from root to L are modified

```
1  X ← L
2  N ← the set of nodes to be removed from the tree // initially, N is empty
3
4  while X is not the root do
5      ParentX ← father node of X
6      EX ← entry of ParentX that corresponds to X
7      if |X| < m entries then
8          Remove EX from ParentX
9          Insert(N, X)
10     end
11     if X not been removed then
12         Adjust its corresponding EX.mbr, so as to enclose all rectangles in X
13     end
14     X ← ParentX
15 end
16 Reinsert all the entries of nodes in N
```

Chapter 5

Experimental Results

5.1 Iris Databases

The proposed indexing approach has been tested on publicly available BATH database [23] and CASIA-IrisV4 [24]. BATH iris database comprises images from 50 subjects. For each subject both left and right iris images are acquired, each containing 20 images of the respective eyes. CASIA-Iris-Thousand subset of CASIA-IrisV4 is used. It contains 20,000 iris images from 1,000 subjects. It is the first publicly available iris dataset with one thousand subjects, and hence, is well-suited for studying the uniqueness of iris features and develop novel iris classification and indexing methods.

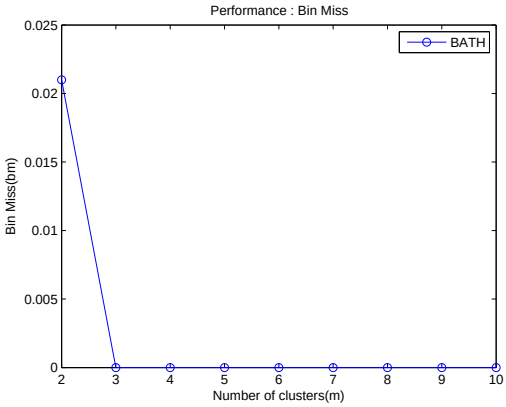
5.2 Experiment

Each cluster centre has 128 elements as feature descriptor, ranges are selected from this to construct a 5-dimensional R-Tree. Each cluster centre is used to construct an R-Tree and a query set is used to get the search in all the clusters. The output is then used to evaluate the performance of the system.

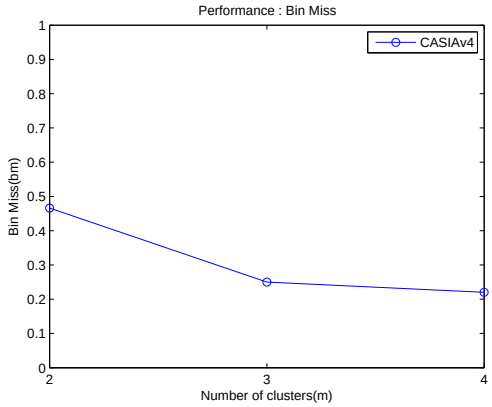
5.3 Performance Measures

Some well known performance measures are used for identification [25].

- **Bin Miss Rate (*bm*):** Bin miss (bm) occurs when probe is wrongly searched in incorrect bin due to indexing errors.



(a) BATH

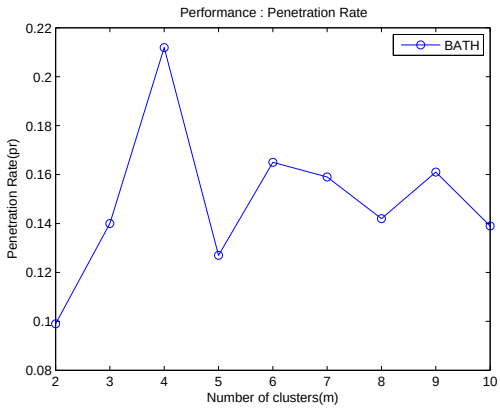


(b) CASIAv4

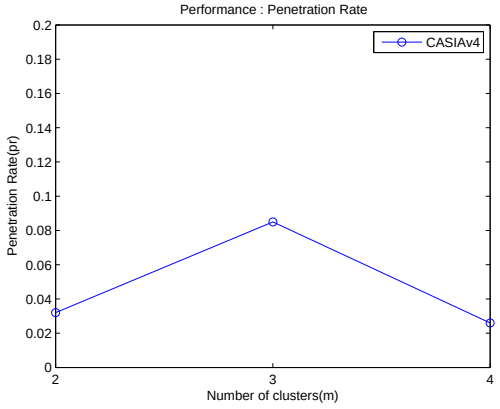
Figure 5.1: Bin miss (bm)

Thus, it is observed from figure 5.1(a), that for m=3 to 10 $bm = 0$ which implies hit rate is 100%, that is the search id is hitting atleast one of the appropriate id in either of the m cluster centres in BATH database. For CASIAv4 figure 5.1(b), hit rate is maximum = 78 % for m=4.

- **Penetration Rate (*pr*):** The penetration rate (pr) of probe search is defined as a ratio of expected number of comparisons against total number of elements in the gallery set.



(a) BATH



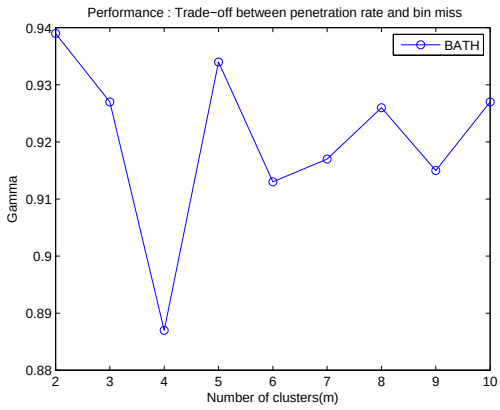
(b) CASIAv4

Figure 5.2: Penetration rate (pr)

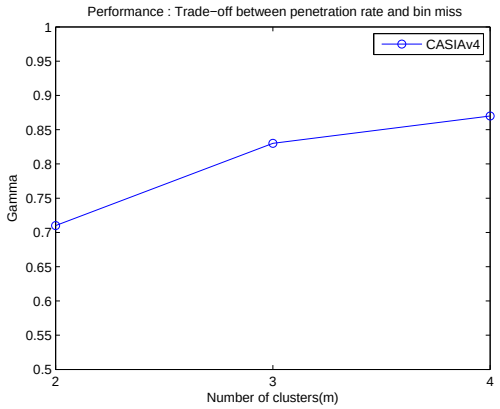
For BATH figure 5.2(a), Penetration Rate is the minimum = 9.9% and 12.7% for m=2 and m=5 respectively, thereby the cost would be reduced. For CASIAv4 figure 5.2(b), penetration is minimum = 2.68% at m=4.

- **Gamma (γ):** The choice of an indexing approach becomes crucial when both speed (measured in terms of pr) and accuracy (measured in terms of bm) plays significant role. Thus, γ [20] is found to strike balance between pr and bm. To mark the trade off between pr and bm a new error measure (γ) is used which is defined as

$$\gamma = \sqrt{(1 - pr) * (1 - bm)} \tag{5.1}$$



(a) BATH



(b) CASIAv4

Figure 5.3: Trade off between pr and bm : γ

The value of m is chosen at optimum point where γ is maximum. For CASIAv4 figure 5.3(b) γ is highest = 87% for m=4. For BATH figure 5.3(a) γ is highest = 93.9 % for m=2. Thus, optimum point is at 2. It is observed that γ comfortably lies between 91% to 94% for most of the values of m, thereby showing an efficient performance of using R-tree as index structure to the Iris database.

5.4 Comparative Analysis

The authors in [5] make use of two approaches for iris indexing. In the first approach,Principal component analysis(PCA) and k-means clustering is used to partition the database. It gives average penetration of 17% for 80% hit rate, whereas the column-based scheme results in a hit rate of 80% for an average penetration rate of 21%. The second approach is based on statistical analysis of pixel intensities and positions of blocks in iris texture. In this case, the average penetration for 80% hit rate is only 8%.

In [20], from the binary iris image, the n-bit pattern is selected and its locations are searched in iris image using burrows wheeler transform (BWT). The experimental results have hit rate of 99.83% at penetration rate of 17.23% and $\gamma = 90.90\%$.

Both the above papers have experiments performed on CASIAv3 database, while in this work, both BATH and CASIAv4 databases are used. For BATH, best performance is seen at m=2 with penetration rate = 9.9%, bin miss = 2.1%, and hence γ = 94%. For CASIAv4, the performance is good at m=4 with penetration rate = 2.6%, bin miss = 22%, and hence γ = 87%. The results highlight improvement in γ over geometric hashing approach. Asymptotically it has been found that R-tree performs retrieval in $O(\log_M n)$ time. This marks the suitability of R-tree based indexing for very large scale iris databases.

Chapter 6

Conclusion

In this thesis, local feature based indexing approach is proposed. R-tree based indexing approach is implemented and tested using publicly available databases. R-tree construction is dynamic and the resultant is a balanced multidimensional index structure, capable of handling the duplicates during insertion. R-Tree has some efficiency problems if there are a lot of unfavourable and multi-dimension records, but it still is a great achievement and opened the door to handling spatial data indexes. In the course of time many new variants of R-Tree were developed to improve the efficiency and thus to improve complex applications. The results obtained mark the suitability of R-tree based indexing for very large scale iris databases.

Bibliography

[1] Unique Identification Authority of India. *http://uidai.gov.in/*.

[2] A. K. Jain and A. Ross and S. Prabhakar. An introduction to biometric recognition, In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 14, pages 4-20, 2004.

[3] A. Mhatre, S. Palla, S. Chikkerur, and V. Govindaraju. Efficient search and retrieval in biometric databases. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 5779, pages 265-273, 2005.

[4] A. Mhatre, S. Chikkerur, and V. Govindaraju. Indexing biometric databases using pyramid technique. In *Audio and Video based Biometric Person Authentication*, pages 841-849, 2005.

[5] R. Mukherjee and A. Ross. Indexing iris images. *19th International Conference on Pattern Recognition*, pages 1-4, 2008.

[6] A. Gyaourova and A. Ross. A coding scheme for indexing multimodal biometric databases. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 93 -98, 2009.

[7] H. Feng, J. Daugman, and P. Zielinski. A fast search algorithm for a large fuzzy database. *IEEE Transactions on Information Forensics and Security*, 3(2):203-212, 2008.

[8] N.B Puhan and N. Sudha. A novel iris database indexing method using the iris color. In *3rd IEEE Conference on Industrial Electronics and Applications*, pages 1886-1891, 2008.

[9] H. Mehrotra, B.G. Srinivas, B. Majhi, and P. Gupta. Indexing Iris Biometric Database using Energy Histogram of DCT Subbands. In *International Conference on Contemporary Computing*, pages 194-204, 2009.

[10] R.B. Gadde, D. Adjeroh, and A. Ross. Indexing iris images using the Burrows-Wheeler Transform. In *IEEE International Workshop on Information Forensics and Security*, pages 1-6, 2010.

[11] U. Jayaraman, S. Prakash, and P. Gupta. An iris retrieval technique based on color and texture. In *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, pages 93-100. ACM, 2010.

[12] C. Rathgeb and A. Uhl. Iris-biometric hash generation for biometric database indexing. In *Proceedings of the International Conference on Pattern Recognition*, pages 2848-2851, 2010.

[13] H. Mehrotra, B. Majhi, and P. Gupta. Robust iris indexing scheme using geometric hashing of SIFT keypoints.*Elsevier Journal of Network and Computer Applications*, 33(3):300-313, 2010.

[14] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf).*Computer Vision and Image Understanding*, 110(3):346 - 359, 2008.

[15] A. Panda, H. Mehrotra, and B. Majhi. Parallel geometric hashing for robust iris indexing. *Journal of Real-Time Image Processing*, pages 1-9, 2011.

[16] D.G. Lowe. Distinctive image features from scale-invariant keypoints.*International Journal of Computer Vision*, 60(2):91-110, 2004.

[17] A. Guttman: R-Trees: A dynamic index structure for spatial searching,*Proceedings of the 1984 ACM SIGMOUND Conference*, S. 47.-57, Boston, MA, 1984

[18] S. Bakshi, H. Mehrotra, and B. Majhi.Real-time iris segmentation based on image morphology. *In Proceedings of the International Conference on Communication, Computing & Security*, pages 335-338. ACM, 2011.

[19] J.C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters.*Journal of Cybernetics*, 3(3):32-57, 1973.

[20] R.B. Gadde, D. Adjeroh, and A. Ross. Indexing iris images using the Burrows-Wheeler Transform. In *IEEE International Workshop on Information Forensics and Security*, pages 1-6, 2010.

[21] R-Trees: Theory and Applications - Yannis Manolopoulos, Alexandros Nanopoulos, Apostolos N. Papadopoulos, Yannis Theodoridis

[22] Sebastian Klabisch R-Tree Proseminar: Algorithms and Datastructures for Database Systems SS 2003

[23] BATH University Database. http://www.bath.ac.uk/elec-eng/research/sipg/irisweb.

[24] CASIA Iris Image Database. http://biometrics.idealtest.org/

[25] J. L. Wayman. Error rate equations for the general biometric system.*IEEE Robotics and Automation Magazine*, 6(1):35-48, 1999.